Chapter 6
Application of Genetic Programming for Electrical Engineering Predictive Modeling: A Review

Seyyed Soheil Sadat Hosseini and Alireza Nemati

6.1 Introduction

Over the last decade, GP has received the interest of streams of researchers around the globe. First, we wanted to provide an outline of the basics of GP, to sum up valuable tasks that gave impetus and direction to research in GP as well as to discuss some interesting applications and directions. Things change fast in this area, as researchers discover new paths of doing things, and new things to do with GP. It is not possible to cover all phases of this field, even within the generous page limits of this chapter.

GP produces computer models to solve a problem utilizing the principle of Darwinian natural selection. GP results are computer programs that are represented as tree structures and shown in a functional programming language (such as LISP) (Koza 1992; Alavi et al. 2011). In other words, programs evolved by genetic programming are parse trees whose length can change throughout the run (Hosseini et al. 2012; Gandomi et al. 2012). GP provides the architecture of the approximation model together with the values of its parameters (Zhang et al. 2011; Gandomi et al. 2011). It optimizes a population of programs based on a fitness landscape specified by a program capability to perform a given task. The fitness of each program is assessed utilized an objective function. Therefore, a fitness function is the objective function that GP optimizes (Gandomi et al. 2010; Javadi and Rezania 2009; Torres et al. 2009). GP and other evolutionary methods have been successfully applied to different supervised learning work like regression (Oltean and Dioan 2009), and unsupervised learning work like clustering (Bezdek et al. 1994; Jie et al. 2004; Falco et al. 2006; Liu et al. 2005; Alhajj and Kaya 2008) and association discovery (Lyman and Lewandowski 2005).

S.S. Sadat Hosseini (✉) • A. Nemati
Department of Electrical Engineering and Computer Science,
University of Toledo, Toledo, OH 43606, USA
e-mail: s.sadathosseini@gmail.com; nemati.alireza@gmail.com

A.H. Gandomi et al. (eds.), *Handbook of Genetic Programming Applications*,
DOI 10.1007/978-3-319-20883-1_6

Our review on the application of GP is focused on electrical engineering, control, optimization and scheduling, signal processing, classification and power system operation. The distinctive features of GP create it a very convenient method in regard to optimization. The application of GP to these areas gives some interesting advantages, the principal one being its flexibility, which lets the algorithm be modified to the needs of each particular problem. In fact, GP typically performs an implicit process of feature selection and extraction. Interpretability can be quickly favored by the utilization of GP since it can utilize more interpretable representation formalism, like rules. It requires to be mentioned that GP approach is an evolutionary method that bears a strong resemblance to genetic algorithm's (GA's). The main differences between GA's and GP can be summed up as follows:

- GP codes solutions as tree structured, variable length chromosomes, but GAs make utilization of chromosomes of fixed length and structure.
- GP usually includes a domain specific syntax that governs meaningful arrangements of information on the chromosome. The chromosomes are syntax free for GAs.
- GP maintains the syntax of its tree-structured chromosomes during 're-production'.
- GP solutions are frequently coded in a way that lets the chromosomes be directly executed utilizing a suitable interpreter. GAs are hardly coded in a directly executable form.

The utilization of this flexible coding system permits the method to carry out structural optimization. This technique can be helpful to the solution of many engineering problems. For instance, GP may be utilized to implement symbolic regression. While conventional regression seeks to optimize the parameters for a pre-specified model architecture with symbolic regression, while the model design and parameters are specified simultaneously. Similarly, the evolution of control methods, Structural design, scheduling programs and signal processing algorithms can be seen as structural optimization problems appropriate for GP. Cramer created one of the first tree structured GAs for primary symbolic regression. Another early development was the BEAGLE technique of Forsyth, which produced classification rules utilizing a tree structured GA. However, it was Koza (1992) who was largely responsible for the popularization of GP within the area of computer science. His GP method (coded in LISP) was applied to a broad range of problems involving symbolic regression, control, robotics, games, classification and power system operation. Engineering applications have started to appear while still dominated by computer scientists. Thus, the objective of this paper is to discuss these recent engineering applications and give an entry point to this quickly expanding areas.

6.2 Genetic Programming

GP is a symbolic optimization method that produces computer programs to solve a problem using the principle of Darwinian natural selection. GP was introduced by Koza as an extension of genetic algorithms (GAs). In GP, a random population of individuals (trees) is created to achieve high diversity. While common optimization techniques represent the potential solutions as numbers (vectors of real numbers), the symbolic optimization algorithms present the potential solutions by structural ordering of several symbols. A population member in GP is a hierarchically structured tree comprising functions and terminals. The functions and terminals are selected from a set of functions and a set of terminals. For example, function set F can contain the basic arithmetic operations ($+$, $-$, $*$, /, etc), Boolean logic functions (AND, OR, NOT, etc.), or any other mathematical functions. The terminal set T contains the arguments for the functions and can consist of numerical constants, logical constants, variables, etc. The functions and terminals are chosen at random and constructed together to form a computer model in a tree-like structure with a root point with branches extending from each function and ending in a terminal. An example of a simple tree representation of a GP model is illustrated in Fig. 6.1.

The creation of the initial population is a blind random search for solutions in the large space of possible solutions. Once a population of models has been created at random, the GP algorithm evaluates the individuals, selects individuals for reproduction, generates new individuals by mutation, crossover, and direct reproduction, and finally creates the new generation in all iterations. During the crossover procedure, a point on a branch of each solution (program) is selected at random and the set of terminals and/or functions from each program are then swapped to create two new programs as can be seen in Fig. 6.2.

The evolutionary process continues by evaluating the fitness of the new population and starting a new round of reproduction and crossover. During this process, the GP algorithm occasionally selects a function or terminal from a model at random

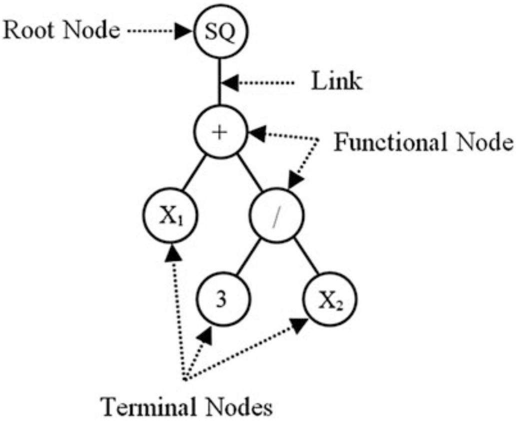


Fig. 6.1 The tree representation of a GP model $(X1 + 3/X2)^2$

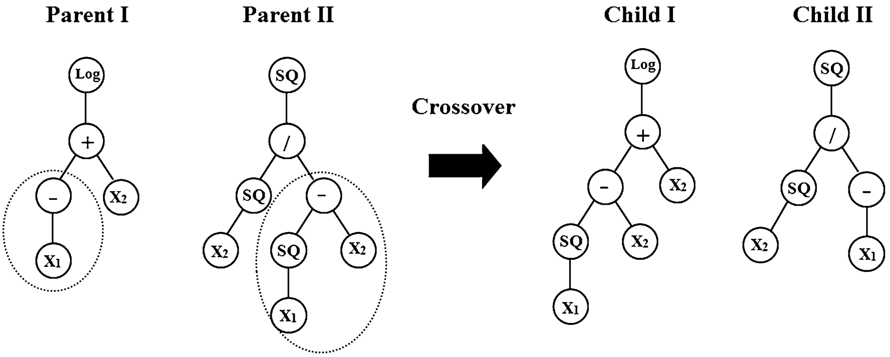
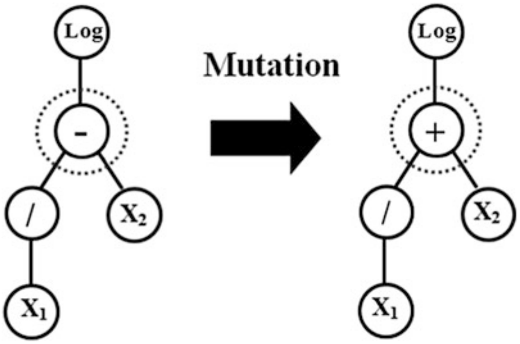


Fig. 6.2 Typical crossover operation in genetic programming

Fig. 6.3 Typical mutation operation in genetic programming



and mutates it (see Fig. 6.3). GEP is a linear variant of GP. The linear variants of GP make a clear distinction between the genotype and the phenotype of an individual. Thus, the individuals are represented as linear strings that are decoded and expressed like nonlinear entities (trees) (Yaghouby et al. 2010; Baykasoglu et al. 2008; Gandomi et al. 2008).

6.3 GP Applications

The following section shows a review of engineering applications of GP. The results of the literature survey have been organized into the following broad groups:

- Control
- Optimization and scheduling
- Signal processing
- Classification
- Power System Operation

6.3.1 Control

Mwaura and Keedwell (2010) used evolutionary algorithms (EAs) to automatically develop robot controllers and occasionally, robot morphology. This field of research is introduced as evolutionary robotics (ER). Through the utilizations of evolutionary methods such as genetic algorithms and genetic programming, ER has proved to be a promising approach through which robust robot controllers can be developed. Ebner (1999) explored the utilization of genetic programming for robot localization to evolve an inverse function mapping sensor readings. This inverse function is defined as an internal model of the environment. Environment is sensed utilizing dense distance information acquired from a laser range finder. An inverse function is developed to localize a robot in a simulated office environment.

Alfaro-Cid et al. (2008) assessed the implementation of genetic programming to design a controller structure. GP is utilized to evolve control strategies that provided the current and desired state of the propulsion and heading dynamics of a supply ship as inputs, produce the commanded forces needed to maneuver the ship. The controllers built utilizing GP are analyzed through real maneuverability tests and computer simulations in a laboratory water basin facility. The robustness of each controller is analyzed through the simulation of environmental disturbances.

Dracopoulos and Kent (1997) emphasized on the application of genetic programming to prediction and control. Results were shown for an oral cancer prediction task and a satellite attitude control problem. Using bulk synchronous model parallelization, the paper explained how the convergence of genetic programming can be significantly speeded up. Nordin and Banzhaf (1997) evaluated the utilization of genetic programming to a direct control a miniature robot. The GP system is employed to evolve real-time obstacle avoiding behavior. Genetic programming enables real-time learning with a real robot. A speed-up of the approach by a factor of more than 2000 was achieved by learning from past. Genetic programming was used in Zell (1999) to search the space of possible programs automatically. First a behavior-based control architecture utilizing computer simulations is evolved. Then one of the experiments with a service robot is replicated, displaying that Kozas classic experiment of evolving a control structure can be transferred to the real world with adjustment to representation. Suwannik and Chongstitvatana (2001) generated the control program by genetic programming to enhance the robustness of a robot arm. The robustness is measured in the real world. To enhance the robustness, multiple robot arm configurations used to evolve the control program. The result showed that the robustness of a control program is enhanced by 10 % in comparison to a control program evolved with a single configuration. Another control related application of GP has been done by Nordin et al. (1997). they have tried to control the khepera robot using GP. Their objective of using GP to control this miniature robot is to evolve real-time obstacle avoiding behavior. Their technique enables real time learning with actual robot. Figures 6.4 and 6.5 show the actual robot and its sensor placement, respectively. The learning that applied to experimental result, had papulation size of 50 individuals. the individuals used values from the sensors as an

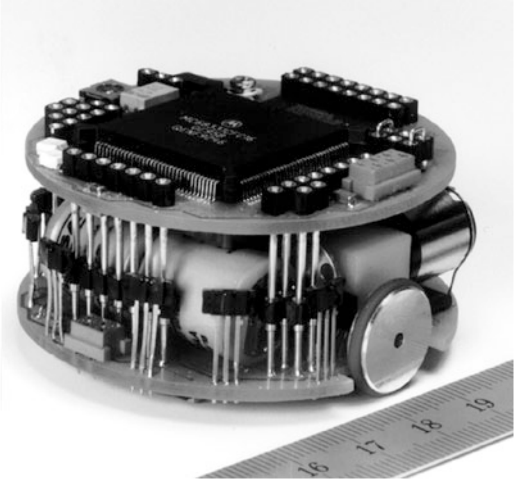


Fig. 6.4 The Khepera robot

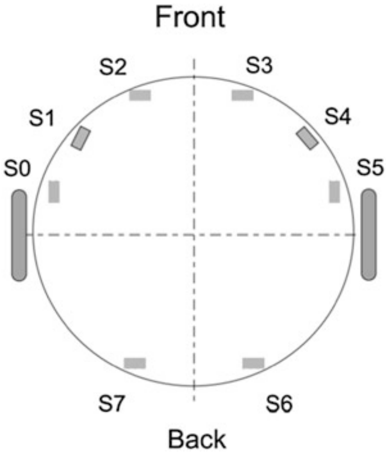


Fig. 6.5 Position of the IR proximity sensors

input and create two outputs values. The output values has been transmitted to the robot as motor speeds. the population of each individuals is processed by the GP system.

Figure 6.6 shows a schematic view of the system. This schematic has been captured from Nordin et al. (1997).

6.3.2 *Optimization and Scheduling*

Grimes (1995) were used genetic algorithm (GA) and genetic programming (GP) methods for track maintenance work with profit as the optimization criteria. The results were compared with an existing method. It was shown that the GP algorithm

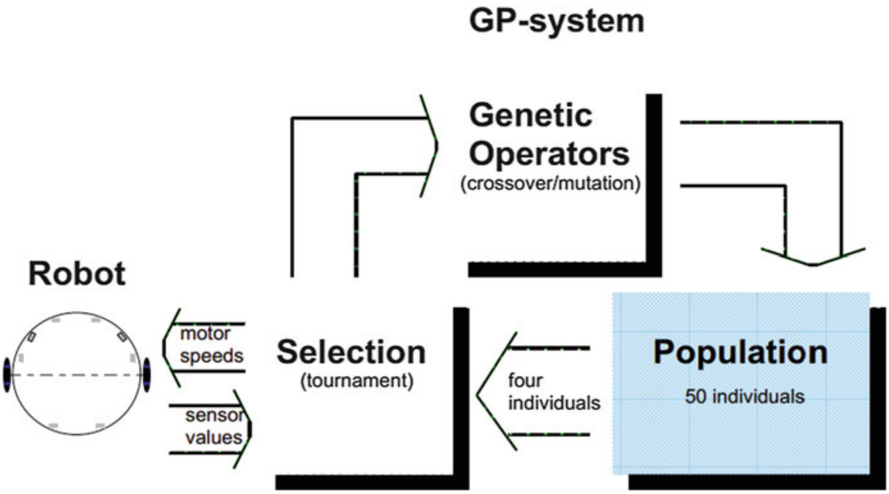


Fig. 6.6 Schematic view of the control system

provided the best results, with the GA approach providing good results for a short section and poor results for a long section of track. Genetic programming was used in Stephenson et al. (2003) to optimize the priority functions associated with two well-known compiler heuristics: predicted hyperblock formation and register allocation. Their system achieved remarkable speedups over a standard baseline for both problems. Vanneschi and Cuccu (2009) presented a new model of genetic programming with variable size population in this paper and applied to the reconstruction of target functions in dynamic environments. This models suitability was tested on a set of benchmarks based on some well-known symbolic regression problems.

Experimental results confirmed that their variable size population model found solutions of similar quality to the ones found by genetic programming, but with a smaller amount of computational effort. Ho et al. (2009) developed an algorithm to derive a distributed method automatically dynamically to optimize the coverage of a femtocell group utilizing genetic programming. The resulting evolved method showed the capability to optimize the coverage well. Also, this algorithm was able to offer increased overall network capacity compared with a fixed coverage femtocell deployment. The evolution of the best-known schedule illustrated in Langdon and Treleaven (1997) for the base South Wales problem utilizing genetic programming starting from the hand coded heuristics. Montana and Czerwinski (1996) applied a hybrid of a genetic algorithm and strongly typed genetic programming (STGP) to the problem of controlling the timings of traffic signals that optimize aggregate performance. STGP learns the single basic decision tree to be executed by all the intersections when determining whether to change the phase of the traffic signal.

6.3.3 Signal Processing

Ahmad and Khan (2012) explored the application of Neuro-Evolutionary Techniques to the diagnosis of various diseases. The evolutionary method of Cartesian Genetic programming Evolved Artificial Neural Network (CGPANN) is applied for the detection of three important diseases. Holladay and Robbins Holladay and Robbins (2007) showed that FIFTH, a new vector-based genetic programming (GP) language, can automatically derive very efficient signal processing techniques directly from signal data. Utilizing symbol rate estimate as an example, the performance of a standard method was compared to an evolved approach. The capabilities of genetic programming were expanded with combining domain knowledge about both machine learning and imaging processing techniques in Harding et al. (2013). The method is shown fast, scalable and robust. A novel genetic programming method was developed in Sharman et al. (1995) to evolve both the parameters and structure of adaptive digital signal processing algorithms. This process is accomplished by determining a set of node terminals and functions to implement the necessary operations commonly utilized in a broad class of DSP techniques. Also, simulated annealing was used to assist the GP in optimizing the numerical parameters of expression trees.

Esparcia Alczar (1998) presented a novel GP approach in the equalization of nonlinear channels. A new way of handling numerical parameters in GP, node gains, was defined. A node gain is a numerical parameter assigned to a node that multiplies its output value. Esparcia-Alczar and Sharman (1999) investigated the application of a combined genetic programming—simulated annealing (GP-SA) solution to a classical signal processing problem. This problem is called channel equalization where the goal is to build a system which adaptively compensates for imperfections in the path from the transmitter to the receiver. Authors were examined the reconstruction of binary data sequences transmitted through distorting channels. Alczar et al. (1996) are also have worked on some application of GP in signal processing in discrete-time manner. They have presented special tree nodes that maintain time recursion, sigmoidal nonlinear transfer functions and internal recursion, which are frequently used operations in signal processing. Table 6.1 which has capture from the same paper, describe these with nodes which implement frequently used algebraic operations in signal processing.

6.3.4 Classification

An algorithm to the utilization of genetic programming was proposed for multiclass image recognition problems in Smart and Zhang (2003). In their method, the terminal set is made with image pixel statistics, the function set includes arithmetic and conditional operators, and the objective function is based on classification precision in the training set. Instead of utilizing fixed static thresholds as boundaries

Table 6.1 The function and terminal node set for evolving discrete-time systems

Symbol	Arty	Description	Symbol	Arty	Description
+,−	2	Addition, subtraction	**xN**	0	System input data. N indicates the delay (e.g. x2 returns xn−2)
*,/	2	Multiplication, division—if second argument is 0, then the node output is set to a large maximum value	**yN**	0	Previous output from the expression tree. The index N indicates 1+ the delay factor (e.g. y2 returns yn−3)
+1,−1	1	Increment, decrement	**z**	1	Unit sample time delay
*2,/2	1	Multiply/divide by two	**fN**	Variable	Execute the **N**th function tree
c**N**	0	Constant value. **N** is an index to a table of constants whose values may be predefined or chosen at random	**argN**	0	The Nth argument to a function tree
nl**N**	1	Non-linear transfer function. **N** indicates the amount of nonlinearity	**psh**	1	Push the argument value onto the stack
avg**N**	N	The average of its **N** arguments	**stkN**	0	Retrieve the Nth item from the stack

to distinguish between different classes, this technique proposed two dynamic algorithms of classification. These methods are centered dynamic range selection and slotted dynamic range selection, based on the returned value of an evolved genetic program where the boundaries between different classes can be dynamically decided during the evolutionary process. GP was applied to solve cost-sensitive classification by means of two techniques through a) manipulating training data and b) adapting the learning method in Li et al. (2005). A constrained genetic programming (CGP), a GP based the cost-sensitive classifier, has been proposed in this paper. CGP is capable of making decision trees to minimize not only the expected number of errors, but also the expected misclassification costs through a novel constraint objective function. The ensemble classification paradigm is an efficient way to enhance the accomplishment and stability of individual predictors. Evolutionary algorithms (EAs) also have been widely utilized to produced ensembles. In the context of heterogeneous ensembles, EAs have been successfully employed to modify weights of base classifiers or to select ensemble members. A novel genetic program was developed in Escalante et al. (2009) that learned a fusion function for integrating heterogeneous-classifiers outputs. It evolves a population of fusion functions to maximize the classification precision. A GP-based method was developed in Liu and Xu (2009) to evaluate multi-class micro-array data sets. In contrast to the standard GP, the individual formulated in this paper includes a set of small-scale ensembles, named as sub-ensemble (indicated by SE). Each SE includes a set of trees. In application, a multi-class problem is split into a set of two-class problems, each of which is addressed by an SE first. The SEs tackling the respective two-class problems are integrated to make a GP individual, so each can address a multi-class problem directly. Efficient algorithms are developed to address the problems arising in the fusion of SEs, and a greedy method is developed to keep high diversity in SEs. Three GP-based methods were proposed in Zhang and Nandi (2007) for addressing multi-class classification problems in roller bearing fault detection. The First method maps all the classes onto the one-dimensional GP output. The second algorithm singles out each class individually by evolving a binary GP for each class independently. The third technique also has one binary GP for each class, but these GPs are evolved together with the goal of choosing as few features as possible. It can also be mentioned that an application of GP in classifiers also could be dividing in three different data set which consist of several subsets. Each of the subsets is also used in an independent run of GP to build up each classifiers. Figure 6.7 shows the classification tasks where GP can be used. This figure has been captured from Zhang and Nandi (2007).

6.3.5 Power System Operation

One of the fundamental power systems planning responsibility of an electrical utility is to precisely anticipate load requirement for all time. The achieved results from load forecasting operation are utilized in various fields like planning and

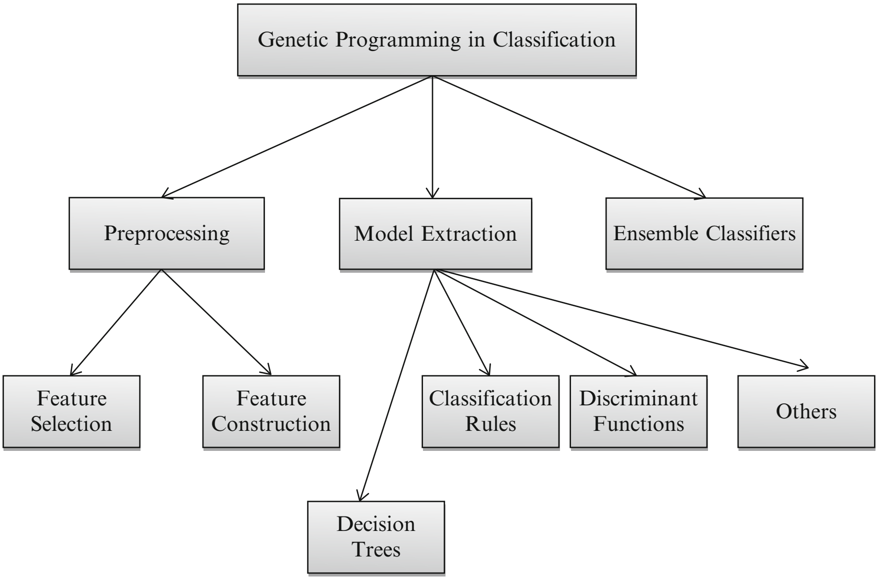


Fig. 6.7 Applications of GP in classification tasks

operation. Preparation of future expenses on construction, rely on the certainty of the long term load foretelling significantly, therefore, various estimation procedure have been tested for short and long term forecasting. Traditional load forecasting approaches are planted on statistical scheme. It needs to be mentioned that the evaluation of load aforetime is usually called as a Load forecasting. This estimation could be demand and energy which is essentially required to improve the system planning effectiveness. These forecasting is also utilized to organized approaches for construction and energy forecast which are essential for future fuel requirements determination. So a good forecast affecting the trend of power planning of present and future. Chaturvedi et al. (1995) are given the GP approach for long term load forecasting.

GP claims to support an optimal solution for the computational problem like power planning. Dr. Kamal (2002) worked on methods of calculation of problems of curve fitting by using GP. He showed that this problem can be carried out without use of equation shape. Farahat (2010) are also applied GP to forecast short term demand by using a new method. Some other researchers are also specified the comparison of different estimation algorithms for power system load forecasting. Genetic pronging, least absolute value filtering and least error squares are different approaches in their experiments. They have considered different forecasting models.

6.4 Conclusions

This paper has provided us with the background context required to understand the reviewed documents and use as a guideline to categorize and sort relevant literature. While computer scientists have focused on gaining a significant understanding of the methods the engineering community is solving practical problems, frequently by introducing accepted systems engineering methodologies and concepts. The combination of different methods permits us to make the most of several algorithms, using their strengths and preventing their drawbacks. The flexibility of GP makes it possible to combine it with very various algorithms. But the combination of GP with some other methods is not the only option; GP can be employed as a mechanism to integrate different techniques. It is stressed that GP is a young area of research, whose practitioners are still exploring its abilities and drawbacks. Therefore, it is the authors' belief that the future holds much promise.

References

Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection. (3rd ed.) MIT press, Cambridge, MA

Alavi AH, Ameri M, Gandomi AH, Mirzahosseini MR (2011) Formulation of ow number of asphalt mixes using a hybrid computational method. Constr Build Mater 25(3):1338–1355

Sadat Hosseini SS, Gandomi AH (2012) Short-term load forecasting of power systems by gene expression programming. Neural Comput Appl 21(2):377–389

Gandomi AH, Alavi AH (2012) A new multi-gene genetic programming approach to non-linear system modeling. Part II: geotechnical and earthquake engineering problems. Neural Comput Appl 21(1):189–201

Zhang Q, Fang J, Wang Z, Shi M (2011) Hybrid genetic simulated annealing algorithm with its application in vehicle routing problem with time windows. Adv Mater Res 148–149:395–398

Gandomi AH, Yang X-S (2011) Benchmark problems in structural optimization. In: Computational Optimization, Methods and Algorithms. Springer, Berlin/Heidelberg, 259–281

Gandomi AH, Alavi AH, Arjmandi P, Aghaeifar A, Seyednour R (2010). Genetic programming and orthogonal least squares: a hybrid approach to modeling the compressive strength of CFRP-confined concrete cylinders. J Mech Mater Struct 5(5):735–753

Javadi A, Rezania M (2009) Applications of artificial intelligence and data mining techniques in soil modeling. Geomech Eng 1(1): 53–74

Torres RS, Falcão AX, Gonçalves MA, Papa JP, Zhang B, Fan W, Fox EA (2009) A genetic programming framework for content-based image retrieval. Pattern Recogn 42:283–92

Oltean M, Diosan L (2009) An autonomous GP-based system for regression and classification problems. Appl Soft Comput 9(1):49–60

Bezdek JC, Boggavarapu S, Hall LO, Bensaid A (1994) Genetic algorithm guided clustering.In: IEEE World Congress on Computational Intelligence

Jie L, Xinbo G, Li-Cheng J (2004) A CSA-based clustering algorithm for large data sets with mixed numeric and categorical values. Fifth World Congress on Intelligent Control and Automation, WCICA, 2303–2307

Falco ID, Tarantino E, Cioppa AD, Fontanella F (2006) An innovative approach to genetic programming-based clustering. In: Proc. 9th Online World Conf. Soft Comput. Ind.

Appl.(Advances in Soft Computing Series,34)., Berlin, Germany: Springer-Verlag, Sep./Oct, 55–64

Liu Y, Ozyer T, Alhajj R, Barker K (2005) Cluster validity analysis of alternative results from multi-objective optimization. In: Proc. 5th SIAM Int Conf Data Mining, Newport Beach, CA, 496–500

Alhajj R, Kaya M (2008) Multi-objective genetic algorithms based automated clustering for fuzzy association rules mining. J Intell Inf Syst 31(3): 243–264

Lyman M, Lewandowski G (2005) Genetic programming for association rules on card sorting data. In: Proc Genet Evol Comput Conf, Washington, DC: ACM, 1551–1552

Yaghouby F, Ayatollahi A, Yaghouby M, Alavi AH (2010) Towards automatic detection of atrial fibrillation: a hybrid computational approach. Comput in Biol Med 40(11–12):919–930

Baykasoglu A, Gullub H, Çanakç H, Özbakir L (2008) Prediction of compressive and tensile strength of limestone via genetic programming. Expert Syst Appl 35(1): 111–123

Gandomi A, Alavi A, Sadat Hosseini S (2008) A Discussion on Genetic programming for retrieving missing information in wave records along the west coast of Indian Applied Ocean Research 2007; 29 (3): 99–111.

Mwaura J, Keedwell E (2010) Evolution of robotic behaviours using Gene Expression Programming. In: IEEE Congress on Evolutionary Computation (CEC), 1–8

Ebner M (1999) Evolving an environment model for robot localization, Euro GP, Ebenhard-Karls-Universitat Tubingen, Germany, Springer Verlag, 184–192

Alfaro-Cid E, McGookin EW, Murray-Smith DJ, Fossen TI (2008) Genetic programming for the automatic design of controllers for a surface ship. IEEE Trans Intell Transp Syst 9(2):311–321

Dracopoulos DC, Kent S (1997) Genetic programming for prediction and control. Neural Comput Appl 6(4):214–228

Nordin P, Banzhaf W (1997) Real time control of a Khepera robot using genetic programmmg. Control Cybern 26(3)

Ebner M, Zell (1999) A Evolving a behavior-based control architecture-From simulations to the real world. In: Proceedings of the Genetic and Evolutionary Computation Conference, 1009–1014

Suwannik W, Chongstitvatana P (2001) Improving the robustness of evolved robot arm control programs with multiple configurations. In: 2nd Asian Symposium on Industrial Automation and Robotics, Bangkok, Thailand

Nordin, Peter, and Wolfgang Banzhaf. “Real time control of a Khepe. ra robot using genetic programmmg.” Control and Cybernetics 26, no. 3 (1997).

Grimes CA, (1995) Application of genetic techniques to the planning of railway track maintenance work. in First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA, IEE: Sheffield, UK, 414, 467–472

Stephenson M, OReilly UM, Martin MC, Amarasinghe S (2003) Genetic programming applied to compiler heuristic optimization, In: Proceedings of the European Conference on Genetic Programming, (Essex, UK), Springer, 238–253

Vanneschi L, Cuccu G (2009) A Study of Genetic Programming Variable Population Size for Dynamic Optimization Problems. In: Proceedings of the International Conference on Evolutionary Computation, part of the International Joint Conference on Computational Intelligence (IJCCI), ed. by A. Rosa et al

Ho, LTW, Ashraf I, Claussen H (2009) Evolving femtocell coverage optimization algorithms using genetic programming. In Personal, Indoor and Mobile Radio Communications, IEEE 20th International Symposium on, 2132–2136

Langdon WB, Treleaven P (1997) Scheduling maintenance of electrical power transmission networks using genetic programming. In KevinWarwick, Arthur Ekwue, and Raj Aggarwal, editors, Artificial Intelligence Techniques in Power Systems, chapter 10, 220–237

Montana DJ, Czerwinski S (1996) Evolving control laws for a network of traffic signals. In: Genetic Programming 1996: Proceedings of the First Annual Conference, Stanford University, CA, USA, 333–338. MIT Press, Cambridge

Ahmad AM, Khan GM (2012) Bio-signal processing using cartesian genetic programming evolved artificial neural network (cgpann). In: Proceedings of the 10th International Conference on Frontiers of Information Technology, 261–268

Holladay K, Robbins K (2007) Evolution of signal processing algorithms using vector based genetic programming. 15th International Conference in Digital Signal Processing, Cardiff, Wales, UK, 503–506

Harding S, Leitner J, Schmidhuber J (2013) Cartesian genetic programming for image processing. In Riolo, R., Vladislavleva, E., Ritchie, M. D., and Moore, J. H., editors, Genetic Programming Theory and Practice X, Genetic and Evolutionary Computation, 31–44. Springer New York

Sharman KC, Alcazar AIE, Li Y (1995) Evolving signal processing algorithms by genetic programming. First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA, IEE, 414, 473–480

Esparcia Alcázar AI (1998) Genetic programming for adaptive digital signal processing. PhD thesis, University of Glasgow, Scotland, UK

Esparcia-Alcázar A, Sharman K (1999) Genetic Programming for channel equalisation. In R. Poli, H. M. Voigt, S. Cagnoni, D. Corne, G. D. Smith, and T. C. Fogarty, editors, Evolutionary Image Analysis, Signal Processing and Telecommunications: First European Workshop, 1596, 126–137, Goteborg, Sweden, Springer-Verlag

Alcázar, Anna I. Esparcia, and Ken C. Sharman. "Some applications of genetic programming in digital signal processing." In Late Breaking Papers at the Genetic Programming 1996 Conference Stanford University, pp. 24–31. 1996

Smart W, Zhang M (2003) Classification strategies for image classification in genetic programming. In: Proceeding of Image and Vision Computing Conference, 402–407, New Zealand

Li J, Li X, Yao X (2005) Cost-sensitive classification with genetic programming. The IEEE Congress on.Evolutionary Computation

Escalante HJ, Acosta-Mendoza N, Morales-Reyes A, Gago-Alonso A (2009) Genetic Programming of Heterogeneous Ensembles for Classification. in Progress in Pattern Recognition, Image Analysis, Computer Vision and Applications, Springer, 9–16

Liu KH, Xu CG (2009) A genetic programming-based approach to the classification of multiclass microarray datasets. Bioinformatics 25(3):331–337

Zhang L, Nandi AK (2007) Fault classification using genetic programming. Mech Syst Signal Pr 21(3):1273–1284

Chaturvedi DK, Mishra RK, Agarwal A (1995) Load Forecasting Using Genetic Algorithms Journal of The Institution of Engineers (India), EL 76, 161–165

Dr. Hanan Ahmad Kamal (2002) Solving Curve Fitting problems using Genetic Programming IEEE MELECON May, 7–9

Farahat MA (2010) A New Approach for Short-Term Load Forecasting Using Curve Fitting Prediction Optimized by Genetic Algorithms 14th International Middle East Power Systems Conference (MEPCON10)19–21